

Modelleren van gebruiksmogelijkheden (use cases)

H.W.M.Gazendam

Faculteit Bestuurskunde, Universiteit Twente

Faculteit Bedrijfskunde, Rijksuniversiteit Groningen

februari 1998

Inhoudsopgave

Inhoudsopgave	1
1. Inleiding	2
1.1. Bedoeling van deze tekst	2
1.2. De rol van verschillende soorten modellen.....	2
2. Organisatie en interactie	4
2.1. Actor	4
2.2. Programma, impuls, actie, activiteit, proces	4
2.3. Organisatie	5
2.4. Zaak, prestatie	6
2.5. Subjectsysteem en objectsysteem	6
2.6. Interactie, overdrachtsactie, communicatieve actie	6
2.7. Algemene structuur van de transactie volgens Dietz.....	7
2.8. Transactie	8
2.9. Transformatie	9
3. De organisatorische use cases	10
3.1. Use cases op interactieniveau	10
3.2. Detaillering van use cases op het niveau van acties	11
3.3. Verdere uitwerking van use cases in scenario's	11
4. De informatiesysteem use cases.....	13
4.1. De gebruiksmogelijkheden (use cases) voor procesuitvoering.....	13
4.2. De gebruiksmogelijkheden (use cases) voor informatiebeheer en systeembeheer	14
4.3. Verbanden tussen use cases	15
Literatuur	17
Bijlage 1. Case Fietsenhandel JEEKAA.....	19

1. Inleiding

1.1. Bedoeling van deze tekst

Het modelleren van gebruiksmogelijkheden (use cases), geïntroduceerd door Jacobson c.s. (1994), is een effectief middel gebleken om de functionaliteit van een systeem vast te leggen. De methode kan niet alleen worden toegepast op informatiesystemen, maar ook op organisaties. Rond het praktisch toepassen van use cases kan echter allerlei verwarring ontstaan, bijvoorbeeld het verwarren van organisaties met informatiesystemen, het door elkaar halen van verschillende beschrijvingsniveaus en het op een onjuiste manier behandelen van relaties tussen use cases. In deze tekst wordt aan de hand van de Case Fietsenhandel JEEKAA (Bijlage 1) uiteengezet hoe een verzameling gebruiksmogelijkheden (use cases) kan op een min of meer systematische wijze top down kan worden gedefinieerd.

1.2. De rol van verschillende soorten modellen

Bij het modelleren gebruiken we verschillende soorten modellen. Waarom gebruiken we eigenlijk niet één samenhangend model? Dat komt omdat we onderscheid willen maken tussen het modelleren van de organisatie (in het bijzonder: het informatie-aspectsysteem van de organisatie) en van het informatiesysteem. Ook willen we onderscheid maken tussen het naar buiten toe zichtbare gedrag van een systeem (de buitenkant) en de interne werking ervan (de binnenkant). Het onderscheid tussen buitenkant en binnenkant is in de informatiekunde gebruikelijk omdat het beter is stapsgewijs te ontwerpen, d.w.z. eerst de eisen aan het informatiesysteem (de z.g. requirements) op te schrijven in de vorm van het te vertonen gedrag, en pas daarna een informatiesysteem te ontwerpen (het z.g. design) dat aan deze eisen voldoet. Eisen en ontwerp, buitenkant en binnenkant worden losgekoppeld. Een voordeel hiervan is dat bij technische vernieuwing de buitenkant in stand kan blijven en alleen de binnenkant hoeft te worden veranderd. Een dergelijk onderscheid vinden we ook in de functionele versus de analytische definitie van een systeem: de functionele definitie specificeert de buitenkant, het naar buiten toe zichtbare gedrag, en de analytische definitie zegt iets over de binnenkant, de interne opbouw van het systeem uit deelsystemen.

Op grond van deze twee onderscheidingen (organisatie/ informatiesysteem en buitenkant/ binnenkant) kunnen we vier soorten modellen afleiden:

1. modellen over de buitenkant van organisaties, zoals use case modellen;
2. modellen over de binnenkant van organisaties, zoals role activity diagrams;
3. modellen over de buitenkant van informatiesystemen, zoals use case modellen, maar dan in een andere rol;
4. modellen over de binnenkant van een informatiesysteem, zoals objectmodellen.

Bij elk van deze modellen kan in principe een model worden gemaakt van de huidige situatie en van een ontworpen, gewenste situatie. Veel oude informatiekundige methoden berusten op het idee dat een huidige organisatorische, niet-geïnformaliseerde, situatie moest worden ondersteund door een nieuw informatiesysteem. Men maakte dan alleen organisatie-modellen van de huidige situatie en informatiesysteemmodellen van de gewenste situatie.

Bij de ontwikkeling van een informatiesysteem maakt men normaliter de verschillende modellen in de volgorde 1, 2, 3, 4. Eerst wordt het relevante deel van de organisatie of het bestuurlijk netwerk in kaart gebracht, eventueel herontworpen, en

daarna wordt de rol en de opbouw van het huidige informatiesysteem (indien aanwezig) beschreven, en vervolgens (her)ontworpen.

Er zijn nog veel andere typen modellen over de binnenkant van een informatiesysteem. Het objectmodel is eigenlijk een statisch model over de bestanddelen van een informatiesysteem. De samenwerking van deze objecten bij het afhandelen van taken kan door diverse typen gedragsmodellen worden beschreven¹.

Opmerkelijk is dat use case modellen in twee rollen voorkomen, namelijk bij de beschrijving van de buitenkant van organisaties en bij de beschrijving van de buitenkant van informatiesystemen. Dit zijn dus verschillende modellen, alhoewel ze wel iets met elkaar te maken hebben, zoals hieronder zal worden uitgelegd.

Organisatorische use cases kunnen op drie niveau's worden beschreven:

- het niveau van de (economische) *transactie* tussen actoren, dit houdt meestal in het leveren van een prestatie en een tegenprestatie, inclusief de nodige voorbereiding en nazorg (contact, contract en controle uit de transactiekostentheorie);
- het niveau van de *interactie* of conversatie tussen actoren, meestal het uiten van een boodschap en een tegenboodschap; soms ook de overdracht van rechten, goederen, diensten of geld gepaard gaande met een verklaring en een aanvaarding (Dietz, 1996);
- het niveau van de *actie* en de *activiteit* die allebei gebeurtenissen zijn die op een bepaald moment plaatsvinden, waarbij een actie geen duur heeft en een activiteit wel.

Informatiesysteem use cases worden vaak op het niveau van de interactie beschreven en soms op het niveau van de actie/ activiteit.

In het boek *The Object Advantage* geven Jacobson c.s. (1994) aan dat de organisatorische use case eigenlijk moet worden beschreven op het niveau van de transactie, omdat dit voor de gebruikende actor een afgerond resultaat geeft. Andere bronnen geven aan dat use cases eerst grof mogen worden gedefinieerd en vervolgens mogen worden verfijnd: een transactie zou dan bijvoorbeeld in termen van interacties kunnen worden uiteengelegd en vervolgens in termen van acties. Om een verband te kunnen leggen tussen organisatorische use cases en informatiesysteem use cases zouden de organisatorische use cases eerst tot op het niveau van de interacties moeten worden uitgeschreven. Vervolgens moet voor een bepaalde actor in een interactie worden bepaald of de rol van deze actor (1) alleen door de mens moet worden uitgevoerd, of (2) door de mens ondersteund door de computer of (3) alleen door de computer (geautomatiseerd dus). Op deze manier kan een verzameling use cases (gebruiksmogelijkheden) voor het informatiesysteem worden afgeleid (d.w.z. die gebruikgevallen waarin het informatiesysteem de mens ondersteunt of vervangt).

2. Organisatie en interactie

2.1. Actor

Een *actor* is een systeem dat zelfstandig handelingen uitvoert gebruik makend van een stelsel van regels, een programma of script, adequaat reagerend op de omstandigheden, zodanig dat daarbij min of meer intelligente beslissingen worden genomen (Gazendam en Jorna, 1993; Carley en Prietula, 1994; Jorna, Gazendam, Heesen en Van Wezel, 1996: 20). Organisaties, mensen en in werking zijnde computers kunnen worden opgevat als actoren. Het beschouwen van mensen als

¹ Zie bijvoorbeeld de website over de Universal Modeling Language (UML): <http://www.rational.com/uml/index.shtml>, of <http://www.rspa.com/spi/OOanal.html>.

actoren betekent overigens niet dat mensen niet meer creatief zouden mogen zijn of mogen improviseren of vage begrippen mogen gebruiken. Het tegendeel is het geval. Het is juist een uitdaging om deze verschijnselen te verklaren door een beroep te doen op goed definieerbare mechanismen in plaats van begrippen met een onduidelijke status. Ook betekent het hanteren van de actorbenadering niet dat machines gelijk of gelijkwaardig zijn aan mensen.

Volgens de *physical symbol system* theorie van Newell en Simon (1972) bestaat een actor (physical symbol system) uit een samenhangend geheel van sensoren, processoren en effectoren dat in staat is tot het zelfstandig toepassen en genereren van symboolstructuren zoals gegevens, kennis, en programma's (Gazendam, 1993: 14; Russell and Norvig, 1995: 31).

Bij actoren moeten we een onderscheid maken tussen actortype en individuele actor. Op grond van de gekozen beschouwingwijze kan elke individuele actor tot één of meer actortype worden gerekend. De actortypen worden onderscheiden op grond van de vervulde rol. Waar we in het navolgende bij de modellering over actoren spreken wordt steeds het *actortype* bedoeld. Daarnaast houden wij ons bij de modellering van organisaties veelal niet bezig met computer-actoren. In de praktijk van de modellering is een actor dus steeds een persoon of een groep van personen die een bepaalde rol vervult. Een persoon kan eventueel worden opgesplitst in de verschillende rollen die hij vervult en komt dan overeen met meerdere actoren. Dit is vooral bij kleine organisaties van belang.

Hoe onderscheiden we rollen? Hierbij is het onderscheiden van interacties (die hieronder worden uitgelegd) behulpzaam. Voor elke interactie is binnen het systeem één bepaalde actor verantwoordelijk. De rol van die actor is het besturen *of* het besturen en het uitvoeren van een bepaalde interactie.

2.2. Programma, impuls, actie, activiteit, proces

Rol, script en programma zijn (in toenemende mate meer formele) beschrijvingen van de handelingen die een actor onder bepaalde omstandigheden normaliter zal doen. Rol, script en programma zijn door een actor uitvoerbare beschrijvingen van handelingen. De meest formele beschrijving is een programma. Een *programma* is een samenhangende beschrijving van een hoeveelheid werk, inclusief eventueel de daarbij behorende doelen, eisen, randvoorwaarden, te volgen procedures en te volgen gedragsregels. Als de actor een computer is, dan is het programma een computerprogramma. Een *taak* is een programma waarvoor de verantwoordelijkheid aan een bepaalde actor is toegewezen. Om een actor een programma te laten uitvoeren is een gebeurtenis nodig die als *impuls* (trigger) werkt.

De uitvoering van een programma leidt tot een feitelijk in de tijd plaatsvindende handelingen. Handelingen kunnen *acties* of *activiteiten* zijn. Om het verschil tussen actie en activiteit uit te leggen hebben we het systeembegrip nodig. Het objectstelsel kan worden beschreven in termen van *toestanden* en *gebeurtenissen* voor. Als een gebeurtenis optreedt hangt de volgende toestand zowel van de gebeurtenis als van de huidige toestand af (Rumbaugh e.a., 1991, p.89).

Een *actie* is een handeling door een actor waarvan de tijdsduur te verwaarlozen valt in het kader van het reticulatieniveau van de toegepaste modellering. Een actie is geassocieerd met een gebeurtenis die als impuls werkt. Een *activiteit* is, in tegenstelling tot een actie, een handeling door een actor die tijd in beslag neemt. In veel gevallen is een activiteit geassocieerd met een toestand waarin het systeem bezig is met het uitvoeren van die activiteit. Bij essentiële acties zoals *overdrachtshandelingen* (overdrachtsacties) of *omzettingshandelingen* wordt veelal

een prestatie geleverd. *Taalhandelingen* (communicatieve acties) bestaan uit op het verzenden en ontvangen van *berichten*.

Een *proces* is een verzameling samenhangende, door impulsrelaties verbonden, acties en activiteiten. Een proces dat op organisatieniveau wordt onderscheiden is een *bedrijfsproces*.

2.3. Organisatie

Een *organisatie* is een samenhangend geheel van (1) samenwerkende actoren, (2) door hen uitgevoerde processen van taakuitvoering en (3) door hen gegenereerde en gebruikte kennis en informatie (Gazendam, 1993: 14; Jorna. Gazendam, Heesen en Van Wezel, 1996: 20).

De grenzen van een organisatie kunnen worden vastgesteld op grond van kenmerken van de samenwerking. Een *werkorganisatie* is een verzameling samenwerkende actoren die wordt onderscheiden op grond van een stabiel patroon van samenwerkingsrelaties. Een organisatie kan ook worden afgebakend op grond van formele relaties tussen personen zoals eigendomsverhoudingen, arbeidscontracten, en dergelijke. In dat geval is een *formele organisatie* een verzameling samenwerkende actoren die wordt onderscheiden op grond van formele relaties tussen actoren. Die formele relaties zijn bijvoorbeeld eigendomsrelaties en contracten zoals arbeidsovereenkomsten. Werkorganisatie en formele organisatie hoeven niet congruent te zijn. Volgens Schmidt (1991) doet men er goed aan om de werkorganisatie als basis te zien en de formele organisatie te interpreteren als een extra laag van gedragsbepalende informatie.

Een organisatie bestaat niet alleen uit menselijke en machine actoren. Mensen en machines kunnen instromen en uitstromen in een organisatie, en toch blijft het karakter van een organisatie daarbij soms min of meer hetzelfde. Om deze reden is het van belang om naast actoren ook de in een organisatie plaatsvindende processen en de symboolstructuren (d.w.z. alle in documenten vastgelegde kennis en informatie) die bij een organisatie horen zoals de programma's van de bedrijfsprocessen, de gedragsregels, de briefhoofden, bij een organisatie te rekenen.

2.4. Zaak, prestatie

Een *zaak* is een voor menselijke beheersing vatbaar stoffelijk object (Algra, Ten Berge en Sleurink, 1986: B1/66).

Een *prestatie* is een handeling die voor degene waarvoor zij wordt verricht waarde heeft.

Prestaties kunnen in het samenspel tussen twee (of meer) organisaties worden geleverd (transactiegerichte prestaties) of binnen een organisatie (transformatiegerichte prestaties).

Een transactiegerichte prestatie kan zijn:

- het in eigendom overdragen van een zaak (P van product);
- het verlenen van een dienst (S van service);
- het verlenen of in eigendom overdragen van een recht (R van rights);
- het in eigendom overdragen van geld (F van finance).

Een *transactie* bestaat in de regel uit een transactiegerichte prestatie en een tegenprestatie. De *tegenprestatie* kan ook nil (N), d.w.z niet aanwezig c.q. gewenst zijn.

Transformatiegerichte prestaties bestaan in de regel uit:

- het creëren of van toestand veranderen van zaken (T van transformation)

Bij het veranderen van de toestand van een zaak kan men denken aan het veranderen van de tijdsdimensie (bewaren, opslaan), van plaats (transporteren) en van hoedanigheid (produceren).

2.5. *Subjectysysteem en objectsysteem*

Het te bestuderen probleemgebied beschouwen wij in termen van een subjectysysteem en een objectsysteem. Het *subjectysysteem* is het geheel van actief handelende actoren, gegroepeerd in systemen die we organisaties noemen. Het *objectsysteem* is het onderwerp van de handelingen der actoren. Het subjectysysteem kan onderdeel van het objectsysteem zijn, d.w.z. de handelende actoren kunnen zichzelf (mede) als onderwerp van handeling hebben (Dietz, 1992: 72).

Het subjectysysteem wordt bestudeerd met behulp van actor-interactiemodellen en bedrijfsprocesmodellen. Het objectsysteem wordt bestudeerd met behulp van objectmodellen.

2.6. *Interactie, overdrachtsactie, communicatieve actie*

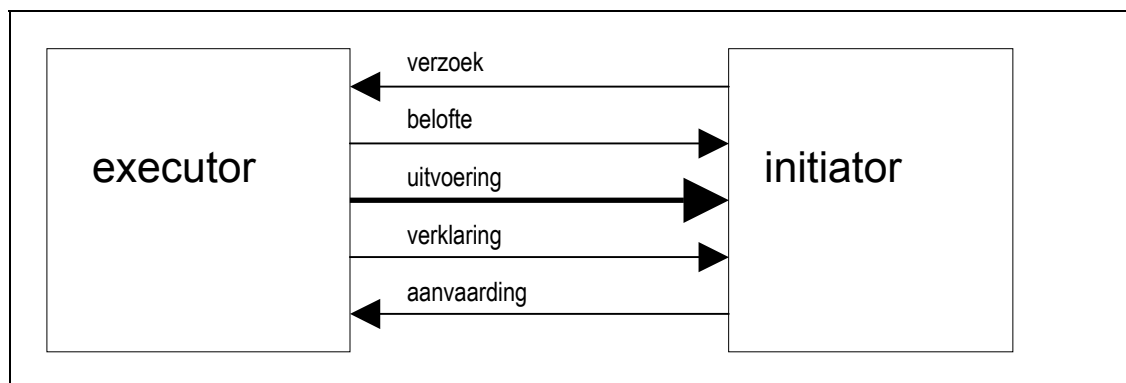
Een *interactie* of *conversatie* is een afgeronde uitwisseling van boodschappen tussen twee actoren, eventueel inclusief een overdracht van rechten, goederen, diensten of geld. Die overdracht van rechten, goederen, diensten of geld is een prestatie (of tegenprestatie).

2.7. *Algemene structuur van de transactie volgens Dietz*

Volgens de theorie van Dietz (1996) heeft een interactie, als we die maar abstract genoeg beschouwen, een standaardstructuur. De kern van een interactie wordt gevormd door een overdrachtshandeling. Om die overdrachtshandeling heen worden berichten uitgewisseld. Bij de interactie zijn twee partijen betrokken: de initiator en de executor. De *executor* voert de overdrachtshandeling uit, terwijl de *initiator* degene is waarvoor de overdrachtshandeling wordt uitgevoerd en die ook het initiatief neemt voor de interactie. Voordat de overdrachtsactie wordt ondernomen, vindt eerst een actagene conversatie plaats waarbij het uitvoeren van de overdrachtsactie op de agenda van de executor wordt geplaatst. Nadat de overdrachtsactie heeft plaatsgevonden moet in een factagene conversatie worden geconstateerd dat zulks het geval is. De hieruit volgende algemene structuur van een interactie is volgens de DEMO methode van Dietz (1996: 33):

- actagene conversatie, bestaande uit de communicatieve acties:
 - verzoek tot uitvoering van de overdrachtsactie door de initiator;
 - belofte tot uitvoering van de overdrachtsactie door de executor;
- uitvoering van de overdrachtsactie (executie) door de executor;
- factagene conversatie, bestaande uit de communicatieve acties:
 - verklaring dat de overdrachtsactie is uitgevoerd door de executor;
 - aanvaarding van de verklaring dat de overdrachtsactie is uitgevoerd door de initiator.

In een *actagene conversatie* komt een agendum tot stand, een punt op het actielijstje



van de uitvoerder van de overdrachtsactie (Dietz, 1992: 78). De actagene conversatie bestaat uit een verzoek en een belofte:

- "Ik verzoek u mij 10 fietsen van het type Gazelle Tour de France Heren te leveren." (*verzoek* door JEEKAA, optredend als Initiator);
- "Ik belof u deze fietsen binnen 12 weken te leveren onder voorwaarde van de bijgesloten leveringsvoorwaarden." (*belofte* door Gazelle, optredend als Executor).

In de *factagene conversatie* komt een feit tot stand, een door beide actoren geaccepteerde constatering (Dietz, 1992: 79).

- "Hierbij lever ik u 10 fietsen van het type Gazelle Tour de France Heren." (*verklaring* van Gazelle, optredend als Executor)
- "OK" (*aanvaarding* van deze verklaring door JEEKAA, optredend als Initiator)

Actagene conversatie en factagene conversatie vormen samen de besturing van de overdrachtsactie. De rol van de initiërende actor is daardoor besturend. De rol van de executerende actor is besturend én uitvoerend.

De hierboven genoemde stappen zijn de stappen die gedaan worden als de interactie op de standaard (default) manier verloopt. De uitzonderingen op dit standaardproces, zoals bijvoorbeeld het intrekken van het verzoek (bijvoorbeeld annulering van een bestelling) kunnen in een toestandsdiagram worden gemodelleerd.

2.8. Transactie

Een *transactie* is een geheel van interacties tussen actoren die tot verschillende organisaties behoren, waarbij op grond van een overeenkomst een uitwisseling van zaken of rechten plaatsvindt. Een transactie is een proces. De meeste gebruikelijke transactie bestaat uit een prestatie en een tegenprestatie, zoals het leveren van een fiets en het betalen van die fiets.

Deze definitie van 'transactie' komt overeen met het algemeen gangbare gebruik van het woord 'transactie'. In de economische transactiekostentheorie (TCE, transaction cost economics) wordt een onderscheid gemaakt tussen drie reticulatieniveau's waarop 'transactie' kan worden gedefinieerd (Williamson, 1985: 1):

- de transactie als overdracht van goed of dienst;
- de transactie als ruilproces waarin de fasen contact, contract en controle worden onderscheiden;
- de transactie als gebeurtenis waarbij rechten van eigendom of beheer worden overgedragen, te onderscheiden van het leveren van de goederen of diensten en het eventueel betalen van die goederen of diensten.

Het meest gedetailleerde reticulatieniveau, namelijk de overdracht van eigendom of beheer, komt overeen met ons interactie-beschrijvingsniveau. Ook het leveren van

goederen of diensten is een interactie, evenals het betalen van deze goederen of diensten. Het meest globale reticulatieniveau van de TCE komt overeen met ons transactie-beschrijvingsniveau. Het ertussen liggende reticulatieniveau geeft ons de aanwijzing om naast contract (het afsluiten van een overeenkomst) en controle (hiertoe zullen we wel de interacties leveren en betalen moeten rekenen) ook een contact-interactie te onderscheiden. We krijgen zo een meer uitgebreide transactiestructuur zoals die bijvoorbeeld door Gale en Eldred (1996: 282) is uitgewerkt:

- opnemen contact (een interactie waarin alleen informatie wordt uitgewisseld, bijvoorbeeld offerte);
- afsluiten overeenkomst c.q. contract (als gevolg van onderhandelen);
- leveren prestatie (meestal: leveren product of dienst);
- leveren tegenprestatie (meestal: betalen);
- verlenen/ verkrijgen nazorg.

In de overheid is het gebruikelijk om de volgende fasen te onderscheiden:

- raming van een uitgave (bijvoorbeeld de jaarlijkse rijksbijdrage aan een universiteit);
- autorisatie om deze uitgave te doen (b.v. doordat een begroting is goedgekeurd);
- verplichting c.q. afsluiten overeenkomst (men heeft een bestelling gedaan of een subsidie toegekend);
- leveren prestatie (er is geleverd of aan andere voorwaarden voldaan en derhalve een recht op betaling ontstaan);
- betaling.

Deze fasen kan men zien als fasen in een transactieproces. Bij dat transactieproces zijn dan meestal drie partijen betrokken, te weten de autoriserende instantie (bijvoorbeeld de Staten-Generaal), de uitvoerende instantie (bijvoorbeeld een minister) en een derde instantie (bijvoorbeeld een universiteit). Er is sprake van een dubbele contractstructuur: er is zowel een fase voor het contract tussen autoriserende instantie en uitvoerende instantie (het goedkeuren van de begroting), als een fase voor het contract tussen uitvoerende instantie en derde instantie (het aangaan van een verplichting).

Meestal komt elke fase in een transactieproces overeen met een *interactie*, d.w.z. uitwisseling van een bericht van de ene actor naar de andere actor, en van een tegenbericht. Soms is een bericht als verklaring gekoppeld aan het leveren van een prestatie. Zo beschouwd, is de standaardstructuur van de interactie van Dietz een uitwerking van twee fasen in het transactieproces, namelijk de fase van het afsluiten van een contract en de fase van het leveren van een prestatie. In wezen bestaat de standaardstructuur van Dietz dus uit *twee* interacties:

- interactie 1: het afsluiten van een overeenkomst c.q. contract:
acties:
 - verzoek (b.v. bestelling) (door initiator);
 - belofte, bevestiging van verzoek (b.v. orderbevestiging) (door executor);
- interactie 2: het leveren van een prestatie:
acties:
 - levering van de prestatie (uitvoering) (door executor);
 - verklaring dat prestatie is geleverd (door executor);
 - aanvaarding van de verklaring dat de prestatie is geleverd (door initiator).

2.9. *Transformatie*

Een transformatie is een geheel van interacties tussen actoren die tot dezelfde organisatie behoren, waarbij zaken worden gecreëerd of van toestand worden veranderd. Waar bij een transactie de centrale plaats wordt ingenomen door overdrachtshandelingen, wordt binnen de transformatie de centrale plaats ingenomen door omzettingshandelingen (omzettingsacties). Een transformatie is een proces. De interacties van een transformatie kunnen eventueel ook zijn tussen een actor en zichzelf. We hebben dan te maken met een zelfactiverend en zelfsturend proces. In het geval van een zelfactiverend, zelfsturend proces kan aan de stappen in de standaard-structuur van de interactie volgens Dietz de volgende betekenis worden toegekend:

Opdrachtfase:

- maken van een plan (verzoek);
- zich vastleggen om het plan uit te voeren (belofte);

Executiefase:

- uitvoeren van de geplande omzettingssactie (uitvoering);
- constateren dat de geplande actie is uitgevoerd (verklaring);

Resultaatafase:

- beoordelen of de uitgevoerde omzettingssactie aan de gestelde eisen voldoet (aanvaarding).

3. De organisatorische use cases

3.1. *Use cases op interactieniveau*

De organisatorische use cases (gebruiksmogelijkheden) beschrijven het gedrag van een organisatie dat waarneembaar is van buiten af, vanuit het gezichtspunt van een actor die met deze organisatie interacteert. Deze use cases kunnen op drie niveau's van detaillering worden beschreven: de transactie, de interactie en de actie.

In het geval van JEEKAA zijn de use cases op transactieniveau:

1. de inkoop bij leveranciers;
2. de verkoop aan klanten;
3. het beantwoorden van vragen over framenummers door derden.

Externe actoren zijn dus de leverancier, de klant en de derde.

Als we deze use cases willen beschrijven op interactieniveau, moeten we kijken naar de producten, diensten, geldhoeveelheden, e.d. die worden uitgewisseld. We zien al gauw dat de inkooptransactie bestaat uit het leveren van goederen door de leverancier en het betalen van de geleverde goederen door JEEKAA. Iets dergelijks geldt ook voor de verkoop. Het beantwoorden van vragen is verder niet op te splitsen, het is een door JEEKAA geleverde dienst zonder tegenprestatie. Verder herinneren wij ons nog dat volgens de transactiekosten economie twee fasen (te interpreteren als interacties) aan de prestatie en tegenprestatie vooraf kunnen gaan, namelijk het contact en het contract. Deze fasen zijn bij de inkoop aanwezig; bij de verkoop is alleen de bestelling aanwezig. Na de prestatie en de tegenprestatie kan er service worden geleverd. Bij de verkoop bestaat deze service o.a. uit het beantwoorden van vragen. Op deze manier krijgen we de volgende organisatorische gebruiksmogelijkheden (use cases):

1. uitbrengen offerte van de leverancier aan JEEKAA (de inkoopofferte);
2. bestellen van JEEKAA bij de leverancier (de inkoopbestelling);

3. leveren van goederen aan JEEKAA door leverancier (de inkooplevering);
4. betalen van geleverde goederen aan de leverancier door JEEKAA (de inkoopbetaling);
5. bestellen van goederen door de klant bij JEEKAA (de verkoopbestelling);
6. leveren van goederen aan de klant door JEEKAA (de verkooplevering);
7. betalen van geleverde goederen aan JEEKAA door de klant (de verkoopbetaling);
8. het beantwoorden van vragen van derden door JEEKAA (de beantwoording).

3.2. Detaillering van use cases op het niveau van acties

Een verdere detaillering van use cases kan van belang zijn voor het detailontwerp van informatiesystemen. Bij een verdere detaillering van deze use cases op het niveau van de acties kijken we naar de berichten die worden uitgewisseld en naar andere niet-talige overdrachten. Elke uitwisseling van goederen, diensten, geld, e.d. kunnen we immers zien als een niet-talige handeling die op gang moet worden gebracht door talige handelingen (berichten zoals brieven, telefoongesprekken), en ook door talige handelingen als afgesloten moet worden verklaard. Voor deze verdere gedetailleerde beschrijving van use cases op het actieniveau is het eerder besproken standaardmodel van Dietz erg handig.

Een voorbeelduitwerking voor JEEKAA is (alleen de inkooptransactie is hieronder uitgewerkt).

Acties:

- 1.1. verzoek om offerte (JEEKAA)
- 1.2. doen van offerte (Leverancier)
- 2.1. Verzoek tot leveren fietsen (JEEKAA)
- 2.2. Belofte tot leveren fietsen (Leverancier)
- 3.1. Uitvoering van leveren fietsen (Leverancier)
- 3.2. Verklaring dat leveren fietsen is uitgevoerd (Leverancier)
- 3.3. Aanvaarding van verklaring dat leveren fietsen is uitgevoerd (JEEKAA)
- 4.1. Verzoek tot betalen fietsen (Leverancier)
- 4.2. Uitvoering van betalen fietsen (JEEKAA)

Bij deze acties worden uitgewisseld (berichten, producten, diensten, geld):

- 1.1. Brief met verzoek om offerte (JEEKAA → Leverancier)
- 1.2. Offerte (Leverancier → JEEKAA)
- 2.1. Bestelformulier (JEEKAA → Leverancier)
- 2.2. Orderacceptatie (Leverancier → JEEKAA)
- 3.1. Fiets (Product) (Leverancier → JEEKAA)
- 3.2. Vrachtbon (Leverancier → JEEKAA)
- 3.3. Aftekenen vrachtbon (JEEKAA → Leverancier)
- 4.1. Inkoopfactuur (Leverancier → JEEKAA)
- 4.2.1. [girale betaling] Bank/giroafschrift (JEEKAA → Bank → Leverancier)
- 4.2.2. [contante betaling] Kwitantie (Leverancier → JEEKAA)

3.3. Verdere uitwerking van use cases in scenario's

Wij gaan voor de verdere behandeling verder met de eerder beschreven acht use cases op interactieniveau. Op grond van andere gegevens (bijvoorbeeld komend uit interviews of bestudering van documenten) kunnen deze use cases verder worden uitgewerkt tot scenario's. Het blijkt bijvoorbeeld dat er drie soorten bestelactiviteiten zijn: de spoedbestelling voor een klant als een fiets niet op voorraad is, de annulering van een bestelling en de reguliere inkoopbestelling. Zo ontstaat uiteindelijk de volgende reeks van use cases.

1. uitbrengen offerte van de leverancier aan JEEKAA (de inkoopofferte);
2. bestellen van JEEKAA bij de leverancier (de inkoopbestelling);
 - 2a. reguliere inkoopbestelling;
 - 2b. spoed-inkoopbestelling;
 - 2c. annulering inkoopbestelling;

3. leveren van goederen aan JEEKAA door leverancier (de inkooplevering);
 4. betalen van geleverde goederen aan de leverancier door JEEKAA (de inkoopbetaling);
 5. bestellen van goederen door de klant bij JEEKAA (de verkoopbestelling);
 6. leveren van goederen aan de klant door JEEKAA (de verkooplevering);
 7. betalen van geleverde goederen aan JEEKAA door de klant (de verkoopbetaling);
 8. het beantwoorden van vragen van derden door JEEKAA (de beantwoording).
- Enkele uitwerkingen van scenario's voor JEEKAA zijn:

- 2a. regulier bestellen fietsen (reguliere inkoopbestelling):
 - opvragen gegevens bestelkosten, voorraadkosten en maken vraagprognose ten behoeve van de bestelformule;
 - uitrekenen te bestellen hoeveelheden met de bestelformule;
 - invullen bestelformulier;
 - verzenden bestelformulier naar leverancier;
- 2b. spoed-inkoopbestelling:
 - opvragen/invullen klantgegevens;
 - kredietwaardigheid nagaan;
 - invullen bestelformulier;
 - verzenden bestelformulier naar leverancier;
- 2c. annuleren inkoopbestelling:
 - opvragen klantgegevens;
 - opvragen bestelgegevens;
 - invullen annulering;
 - verzenden annulering naar leverancier;
3. arriveren fietsen (inkooplevering):
 - opvragen bestelgegevens;
 - invoeren fietsgegevens;
4. inkoopbetaling:
 - opvragen wekelijkse bestelgegevens, fietsgegevens en verkoopgegevens;
 - verzenden wekelijkse gegevens naar financiële administratie.
5. verkopen fietsen (verkoopbestelling):
 - opvragen/invullen klantgegevens;
 - kredietwaardigheid nagaan;
 - opvragen fietsgegevens;
 - invullen verkoopformulier;
 - verzenden verkoopformulier naar financiële administratie;
8. beantwoorden vraag:
 - opvragen gegevens fietsen;

4. De informatiesysteem use cases

4.1. De gebruiksmogelijkheden (use cases) voor procesuitvoering

De gewenste gebruiksmogelijkheden (use cases) van een informatiesysteem worden in drie stappen afgeleid uit de organisatorische use cases op interactieniveau:

1. selecteren van die organisatorische use cases die het informatiesysteem moet ondersteunen of automatisch uitvoeren (dit zijn de use cases voor procesuitvoering);
2. toevoegen van use cases voor de ondersteuning van informatiebeheer;

3. toevoegen van use cases voor de ondersteuning van systeembeheer.
Voor de JEEKAA case gaat dit als volgt.

De oorspronkelijke organisatorische use cases waren:

1. uitbrengen offerte van de leverancier aan JEEKAA (de inkoopofferte);
2. bestellen van JEEKAA bij de leverancier (de inkoopbestelling);
 - 2a. reguliere inkoopbestelling;
 - 2b. spoed-inkoopbestelling;
 - 2c. annulering inkoopbestelling;
3. leveren van goederen aan JEEKAA door leverancier (de inkooplevering);
4. betalen van geleverde goederen aan de leverancier door JEEKAA (de inkoopbetaling);
5. bestellen van goederen door de klant bij JEEKAA (de verkoopbestelling);
6. leveren van goederen aan de klant door JEEKAA (de verkooplevering);
7. betalen van geleverde goederen aan JEEKAA door de klant (de verkoopbetaling);
8. het beantwoorden van vragen van derden door JEEKAA (de beantwoording).

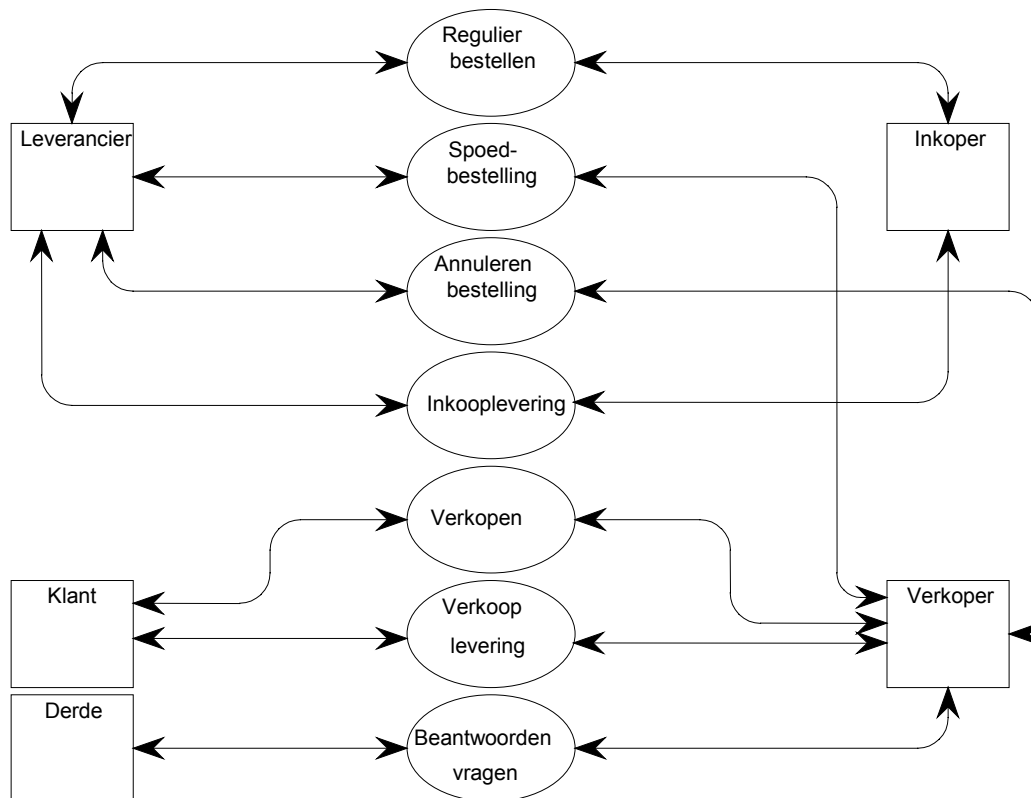
Voor het fietsinformatiesysteem wordt besloten dat dit zich niet bezig gaat houden met financiële zaken. Dit betekent dat de use cases 4 en 7 niet door het fietsinformatiesysteem zullen worden ondersteund. Ook wordt er van afgezien om de offertes (use case 2) via het informatiesysteem te registreren. Dit leidt tot de volgende use cases voor het fietsinformatiesysteem m.b.t. de procesuitvoering.

Procesuitvoering:

(opnieuw genummerd!)

1. reguliere inkoopbestelling door JEEKAA bij de leverancier (regulier bestellen);
2. spoed-inkoopbestelling door JEEKAA bij de leverancier (spoedbestelling);
3. annulering inkoopbestelling door JEEKAA bij de leverancier (annuleren bestelling);
4. leveren van goederen aan JEEKAA door leverancier (inkooplevering);
5. bestellen van goederen door de klant bij JEEKAA (verkopen);
6. leveren van goederen aan de klant door JEEKAA (verkooplevering);
7. het beantwoorden van vragen van derden door JEEKAA (beantwoorden vragen).

Deze use cases kunnen als volgt worden afgebeeld.



4.2. De gebruiksmogelijkheden (use cases) voor informatiebeheer en systeembeheer

Hieraan worden toegevoegd de gebruiksmogelijkheden (use cases) van het informatiesysteem voor informatiebeheer en systeembeheer:

Een informatiesysteem ondersteunt taken altijd op grond van gegevens die over objecten in de wereld zijn geregistreerd. Het *registreren van objecten* dicht bij de bron is dan ook een belangrijke taak die door informatiesystemen wordt ondersteund. Met name het up-to-date houden van gegevens (over b.v. actoren en typen) die niet gekoppeld zijn aan te ondersteunen gebeurtenissen is een apart te onderscheiden taak. Het toewijzen van het beheer van objecttypen is overigens een belangrijke beslissing. De bedoeling is dat degene die het meeste met bepaalde objecttypen werkt en de gegevens over de er onder vallende objecten ook in de meeste gevallen registreert de verantwoordelijkheid voor die objecttypen krijgt. Daarnaast zijn er redeneerregels nodig voor het informatiesysteem om bepaalde gevolgtrekkingen te kunnen maken en bepaalde simulaties of rapportages te kunnen verrichten. Deze redeneerregels (eventueel wettelijke regels) moeten ook zoveel mogelijk dicht bij de bron ervan worden geregistreerd. Soms zijn deze regels gebaseerd op expliciete of geëliciteerde kennis die door een knowledge engineer wordt onderhouden. Tenslotte is er werk nodig om het informatiesysteem zelf te onderhouden en in goede staat te houden: het *systeembeheer*. Op het gebied van systeembeheer zijn er meestal de algemene taken van het maken van back-ups, het opschonen van bestanden (o.a. door het wegschrijven van niet meer gebruikte gegevens in historische bestanden) en het beveiligen van de toegang door passwords e.d.

Een en ander leidt voor JEEKAA tot de volgende gebruiksmogelijkheden voor informatiebeheer en systeembeheer.

Informatiebeheer:

8. klantgegevens beheren (actor: verkoper; trigger: verhuisbericht klant of ander bericht over klant komt binnen);
9. algemene fietsgegevens beheren (actor: inkoper; trigger: nieuwe catalogus komt binnen);
10. leveranciersgegevens beheren (actor: inkoper; trigger: verhuisbericht, naamsverandering, of ander bericht leverancier komt binnen);
11. bijwerken algemene gegevens (actor: systeembeheerder; trigger: organisatieverandering, verhuizing of ander relevante gebeurtenis).
12. inkoopformule en verwante regels beheren (actor: inkoper; trigger: veranderde inzichten).

Systeembeheer:

13. maken backup (actor: systeembeheerder; trigger: het is 8.00 uur);
14. opschonen bestanden (actor: systeembeheerder; trigger: het is 1 januari);
15. beveiliging (actor: systeembeheerder; trigger: het is de eerste van de maand, diversen).

Een en ander leidt dus tot 15 gebruiksmogelijkheden (use cases) van het fietsinformatiesysteem, die samen bepalen wat het fietsinformatiesysteem geacht wordt te doen in de organisatie.

4.3. Verbanden tussen use cases

Met name voor de use cases voor procesuitvoering is het van belang mogelijke verbanden tussen use cases te onderzoeken. Die verbanden worden onderscheiden op het aanwezig zijn van gemeenschappelijke elementen in de afhandeling van use cases. Het onderkennen van gemeenschappelijke elementen in de afhandeling maakt het mogelijk om de afhandeling daarvan door het informatiesysteem slechts éénmaal in te programmeren in plaats van meerdere malen. Verder kan soms tot een meer logische afhandeling van taken worden besloten op grond van verbanden tussen use cases. Bij het onderzoeken van verbanden tussen use cases zijn beschrijvingen van scenario's het uitgangspunt.

Uit de voor JEEKAA gemaakte scenario's blijkt dat het regulier bestellen en de spoedbestelling gemeen hebben dat er een bestelformulier wordt ingevuld en verzonden naar de leverancier. Deze gemeenschappelijke activiteit noemen we 'bestellen'. De aan het 'bestellen' voorafgaande activiteiten bij het regulier bestellen noemen we 'toepassen bestelformule'. De aan het 'bestellen' voorafgaande activiteiten bij de spoedbestelling noemen we 'voortraject spoedbestelling'. We kunnen dan noteren als verbanden tussen use cases:

- 'regulier bestellen' USES 'toepassen bestelformule';
- 'regulier bestellen' USES 'bestellen';
- 'spoedbestelling' USES 'voortraject spoedbestelling';
- 'spoedbestelling' USES 'bestellen'.

Het 'bestellen' zonder voortraject komt incidenteel ook voor. In dat geval kunnen we het 'toepassen bestelformule' en het 'voortraject spoedbestelling' zien als optionele uitbreidingen van het 'bestellen'. We kunnen dus opschrijven:

- 'toepassen bestelformule' EXTENDS 'bestellen';
- 'voortraject spoedbestelling' EXTENDS 'bestellen'.

We zien dat de formule die EXTENDS gebruikt korter is dan dezelfde formule die USES gebruikt. Waar dat zonder de herkenbaarheid voor de gebruiker van het informatiesysteem aan te tasten kan, zouden dus de meer gecompliceerde use cases die door USES zijn samengesteld kunnen vervallen ten faveure van een EXTENDS constructie. Bij JEEKAA zou dat betekenen dat de use cases 'regulier' bestellen en 'spoedbestelling' zouden vervallen. Dit doen we echter bij JEEKAA niet vanwege het verlies aan herkenbaarheid voor de toekomstige gebruiker.

Het annuleren van een bestelling kunnen we zien als een optionele uitbreiding van het 'bestellen'. Het komt dus ook voor bij het regulier bestellen en de spoedbestelling. We schrijven daarom op:

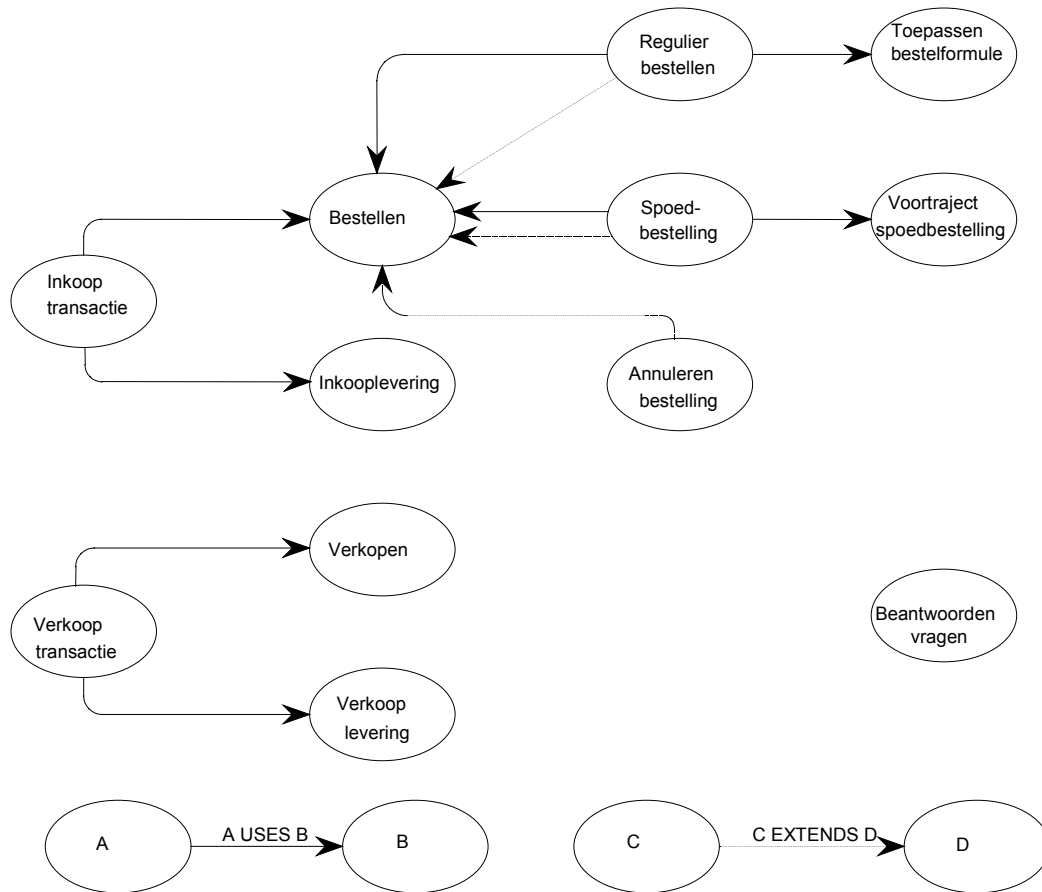
- 'annuleren bestelling' EXTENDS 'bestellen'.

Verder kunnen we het hele proces van inkoop (de inkooptransactie) als één use case op een hoger niveau zien, evenals het hele proces van verkoop (de verkooptransactie). Deze transacties bestaan uit een aantal kleinere use cases. We schrijven op:

- 'inkooptransactie' USES 'bestellen';
- 'inkooptransactie' USES 'inkooplevering';
- 'verkooptransactie' USES 'verkopen';
- 'verkooptransactie' USES 'verkooplevering'.

Of men het beantwoorden van vragen als een service ziet die onderdeel is van de verkooptransactie is een kwestie van smaak. Wij rekenen het beantwoorden van vragen niet onder de verkooptransactie.

Hieronder zijn deze verbanden tussen use cases schematisch weergegeven. Daarbij zijn de gewone, getrokken pijlen de USES relaties en de gestippelde pijlen de EXTENDS relaties.



Literatuur

- Algra, N.E., J.B.J.M. ten Berge en P.H. Sleurink. *Poly-Juridisch Zakboekje*. Arnhem: PBNA, 1986.
- Carley, K.M., and M.J.Prietula (eds.). *Computational Organization Theory*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1994.
- Dietz, J.L.G. *Introductie tot DEMO: Van informatietechnologie naar organisatietechnologie*. Alphen a/d Rijn: Samsom, 1996.
- Dietz, J.L.G. *Leerboek Informatiekundige Analyse*. Deventer: Kluwer Bedrijfswetenschappen, 1992.
- Gale, Thornton, and James Eldred. *Getting Results with the Object-Oriented Enterprise Model*. New York: SIGS, 1996.
- Gazendam Henk W.M. *Variety Controls Variety: On the Use of Organization Theories in Information Management*. Groningen: Wolters-Noordhoff, 1993.
- Gazendam, Henk W.M., and René J.Jorna. "Theories about Architecture and Performance of Multi-Agent Systems." *Paper presented at the III European Congress of Psychology, July 4-9-1993, Tampere, Finland*.
- Jacobson, Ivar, Maria Ericsson, Agneta Jacobson. *The Object Advantage: Business Process Reengineering with Object Technology*. Wokingham, England: Addison-Wesley, 1994.
- Jorna, R.J., H.W.M.Gazendam, H.C.Heesen en W.M.C.van Wezel. *Plannen en Roosteren: Taakgericht analyseren, ontwerpen en ondersteunen*. Leiderdorp: Lansa, 1996.
- Newell, A., and H.A.Simon. *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall, 1972.
- Rumbaugh, James, Michael Blaha, William Premerlani, Frederick Eddy and William Lorensen. *Object-Oriented Modeling and Design*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- Russell, Stuart J., and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice-Hall, 1995.
- Schmidt, K."Cooperative work: conceptual framework." In: J. Rasmussen, B. Brehmer and J.Leplat (eds.). *Distributive decision making: cognitive models for cooperative work*. Chichester: Wiley, 1991.
- Williamson, Oliver E. *The Economic Institutions of Capitalism*. New York: The Free Press, 1985.

Bijlage 1. Case Fietsenhandel JEEKAA

Omschrijving

Fietsenhandel JEEKAA, opgericht door Jeen Kaats, bestaat al 22 jaar. De laatste jaren is het bedrijf door toenemende vraag gestaag gegroeid. Het is een typische eenmanszaak. Er werken naast de eigenaar nog twee verkopers en drie onderhoudsmonteurs. Voor de koopavond en de zaterdag is er nog een parttime verkoper. De laatste tijd heeft de eigenaar het gevoel dat hij het overzicht wat verliest. Om zijn voorraadadministratie doorzichtelijker te maken verzoekt de eigenaar U een oplossing te zoeken. Hieronder staan enkele details van de gang van zaken in het bedrijf.

De activiteiten die betrekking hebben op de voorraad vallen uiteen in drie categorieën. Het bestellen van de fietsen, het arriveren van bestelde fietsen en het verkopen van fietsen. Deze drie processen worden nu handmatig uitgevoerd.

Het bestellen van fietsen gebeurt met behulp van bestelformulieren (zie bijlage 1). Het bestelmoment is van een aantal factoren afhankelijk. Over het algemeen worden de bestelformulieren ingevuld op een moment waarop de eigenaar toevallig even wat tijd heeft. De bestellingen gebeuren grotendeels op het gevoel. Er bestaat geen duidelijk inzicht in de aanwezige voorraad. Soms vraagt een klant om een specifieke fiets die niet in voorraad is. Deze fiets wordt dan op dat moment genoteerd om besteld te worden. Kopieën van de bestelformulieren worden bewaard in een ordner.

Als bestelde fietsen binnenkomen, wordt hiervan de vrachtbon bewaard in een ordner. De factuur, die bij aflevering wordt meegegeven of later wordt opgestuurd, wordt ook in een ordner opgeborgen. Bij binnenkomst van een bestelde fiets moet in principe hiervan een aantekening op het bijbehorende bestelformulier komen. In de praktijk gebeurt dit echter nooit. Het terugzoeken van de juiste bestelformulier is tijdrovend, en bovendien vinden de werknemers het zinloos. Na het binnenkomen van de fiets wordt deze door een technicus uitgepakt en gecontroleerd. Vervolgens worden, indien de klant dat heeft besteld, bepaalde speciale onderdelen zoals andere zadels, bagagedragers, drinkflessen, fietstassen, sloten gemonteerd.

Bij verkoop van een fiets worden de gegevens van de fiets en de klantgegevens handmatig ingevuld op een voorgedrukt formulier. Dit formulier dient als verkoopfactuur. Een doorslag wordt gebruikt voor de administratie van de winkel. Een tweede doorslag van dit formulier wordt eventueel voor de verzekering gebruikt. Deze verzekering betreft voor brommers de WA-verzekering en voor fietsen en brommers een diefstalverzekering. Nadat een fiets is verkocht wordt de fiets door een technicus voor aflevering klaar gemaakt door onder meer schoonmaken, in de was zetten, oliën en banden oppompen.

Het gebeurt steeds vaker dat er gegevens moeten worden opgezocht. Het komt bijvoorbeeld voor dat een bestelling geannuleerd wordt. Dit moet op het bijbehorende bestelformulier worden vermeld. Het terugvinden van dit formulier is lastig. Er is geen overzicht over de aanwezige voorraad. Het komt regelmatig voor dat er om een specifieke fiets wordt gevraagd. De voorraad is echter gegroeid tot zo'n 400 fietsen, waardoor het moeilijk is om een specifieke fiets terug te vinden. Hier komt nog bij dat de eigenaar nu niet precies weet wat hij moet bestellen. Dit gebeurt nu min of meer intuïtief. Wanneer er vragen binnenkomen over een verkochte fiets zijn de gegevens van deze fiets moeilijk terug te vinden. Dit gebeurt vooral in het geval van diefstal: een klant wil het framenummer van zijn fiets weten, of de politie wil van een gevonden fiets de eigenaar achterhalen.

Bestelformulier

BESTELFORMULIER				
Volgnummer	<input style="width: 90%;" type="text"/>			
JEEKAA Tweewielers Kooistraat 12 3354 FF Schiermonnikoog Kamer van Koophandel Groningen 3251				
Datum	<input style="width: 100%;" type="text"/>			
Aan:	Leveranciersnr:	<input style="width: 100%;" type="text"/>		
Naam	<input style="width: 100%;" type="text"/>			
Adres	<input style="width: 100%;" type="text"/>			
Postcode	<input style="width: 100%;" type="text"/>			
Woonplaats	<input style="width: 100%;" type="text"/>			
Hierbij plaats ik de volgende bestelling:				
Type	Soort	Hoogte	Kleur	Aantal
Hoogachtend, Jeen Kaats, directeur				

Factuur

FACTUUR						
Factuurnr						
Batalle Fietsenfabriek Groeneweg 27 8233 AF Beesten						
Datum						
Klantnr	441					
Naam	JEEKAA					
Adres	Kooistraat 12					
Postcode	NL-3354 FF					
Woonplaats	Schiermonnikoog					
Type	Soort	Hoogte	Kleur	Framenr	Prijs	Korting
Totaal						
BTW 17,5%						
Factuurbedrag						
Betaling binnen 14 dagen. Kamer van Koophandel Peerstekel nr.3456						